

WHAT IS CLAIMED IS:

1. A system for processing operations that use data vectors each comprising a plurality of data elements comprising:

5 a vector data file comprising a plurality of storage elements for storing data elements of the data vectors;

 a pointer array coupled by a bus to the vector data file, the pointer array including a plurality of
10 entries wherein each entry identifies at least one storage element in the vector data file; and

 the at least one storage element for storing at least one data element of the data vectors, wherein for at least one particular entry in the pointer array, the at
15 least one storage element identified by the particular entry has an arbitrary starting address in the vector data file.

2. The system as recited in claim 1, wherein, for any given entry in the pointer array, the at least one storage
20 element identified by the any given entry includes an arbitrary starting address in the vector data file.

3. The system as recited in claim 1, wherein the pointer array includes at least one entry which is updated based on data read out from at least one data element in the vector data file.

5

4. The system as recited in claim 1, wherein the pointer array includes at least one entry which is updated based on data generated by incrementing data read from at least one entry of the pointer array.

10

5. The system as recited in claim 1, wherein the pointer array includes at least one entry which is updated based on data generated by performing an increment operation on data read from at least one entry of the pointer array.

15

6. The system as recited in claim 5, wherein the pointer array includes at least two entries which are updated as part of a same logical operation.

20

7. The system as recited in claim 5, wherein the increment operation includes at least one of a modulo operation and a stride operation.

8. The system as recited in claim 1, wherein each entry of the pointer array includes a starting address of at least one storage element in the vector data file.

5 9. The system as recited in claim 1, wherein the storage elements of the vector data file are logically organized in a matrix of rows and columns, and wherein each entry of the pointer array includes an address representing the row and column of at least one element in the vector
10 data file.

10. The system as recited in claim 1, wherein the storage elements of the vector file data are logically organized in a matrix of rows and columns, and wherein each
15 array of the pointer array includes an address representing the row and column of a single element in the vector data file.

11. The system as recited in claim 1, wherein, for any
20 given entry in the pointer array, the at least one storage element identified by the any given entry is independent

with respect to the at least one storage element identified by other entries of the pointer array.

12. A method for processing operations that use data
5 vectors each comprising a plurality of data elements, the method comprising the steps of:

providing a vector data file comprising a plurality of storage elements for storing data elements of the data vectors, and

10 providing a pointer array having a plurality of entries, wherein each entry identifies at least one storage element in the vector data file for storing at least one data element of the data vectors, wherein for at least one particular entry in the pointer array, the at least one
15 storage element identified by the particular entry has an arbitrary starting address in the vector data file.

13. The method as recited in claim 12, wherein, for any given entry in the pointer array, the at least one
20 storage element identified by the any given entry has an arbitrary starting address in the vector data file.

14. The method as recited in claim 12, further comprising the step of:

updating at least one of the entries of the pointer array based on data read out from at least one data element in the vector data file.

15. The method as recited in claim 12, further comprising the step of:

updating at least one of the entries of the pointer array based on data read out from data generated by incrementing data read from at least one entry of the pointer array.

16. The method as recited in claim 12, further comprising the step of:

updating at least one of the entries of the pointer array based on data generated by performing an increment operation on data read from at least one entry of the pointer array.

17. The method as recited in claim 16, wherein at least two entries of the pointer array are updated as part of a same logical operation.

5 18. The method as recited in claim 16, wherein the increment operation further includes at least one of a modulo operation and a stride operation on data read from at least one entry of the pointer array.

10 19. The method as recited in claim 12, wherein each entry of the pointer array stores a starting address of at least one storage element in the vector data file.

15 20. The method as recited in claim 12, wherein the storage elements of the vector data file are logically organized in a matrix of rows and columns, and wherein each entry of the pointer array stores an address representing the row and column of at least one element in the vector data file.

20 21. The method as recited in claim 12, wherein the storage elements of the vector file data are logically

organized in a matrix of rows and columns, and wherein each array of the pointer array stores an address representing the row and column of a single element in the vector data file.

5

22. The method as recited in claim 12, wherein, for any given entry in the pointer array, the at least one storage element identified by the any given entry is independent with respect to the at least one storage element identified by other entries of the pointer array.

10

23. A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for processing operations that use data vectors each comprising a plurality of data elements, the method steps comprising:

15

providing a vector data file comprising a plurality of storage elements for storing data elements of the data vectors, and

20

providing a pointer array having a plurality of entries, wherein each entry identifies at least one storage element in the vector data file for storing at least one

data element of the data vectors, wherein for at least one particular entry in the pointer array, the at least one storage element identified by the particular entry has an arbitrary starting address in the vector data file.

5

24. The program storage device as recited in claim 23, wherein, for any given entry in the pointer array, the at least one storage element identified by the any given entry has an arbitrary starting address in the vector data file.

10

25. The program storage device as recited in claim 23, further comprising the step of:

updating at least one of the entries of the pointer array based on data read out from at least one data element in the vector data file.

15

26. The program storage device as recited in claim 23, further comprising the step of:

updating at least one of the entries of the pointer array based on data read out from data generated by incrementing data read from at least one entry of the pointer array.

20

27. The program storage device as recited in claim 23,
further comprising the step of:

updating at least one of the entries of the
pointer array based on data generated by performing an
5 increment operation on data read from at least one entry of
the pointer array.

28. The program storage device as recited in claim 27,
wherein at least two entries of the pointer array are
10 updated as part of a same logical operation.

29. The program storage device as recited in claim 27,
wherein the increment operation further includes at least
one of a modulo operation and a stride operation on data
15 read from at least one entry of the pointer array.

30. The program storage device as recited in claim 23,
wherein each entry of the pointer array stores a starting
address of at least one storage element in the vector data
20 file.

31. The program storage device as recited in claim 23,
wherein the storage elements of the vector data file are
logically organized in a matrix of rows and columns, and
wherein each entry of the pointer array stores an address
5 representing the row and column of at least one element in
the vector data file.

32. The program storage device as recited in claim 23,
wherein the storage elements of the vector file data are
10 logically organized in a matrix of rows and columns, and
wherein each array of the pointer array stores an address
representing the row and column of a single element in the
vector data file.

15 33. The program storage device as recited in claim 23,
wherein, for any given entry in the pointer array, the at
least one storage element identified by the any given entry
is independent with respect to the at least one storage
element identified by other entries of the pointer array.

34. A method for processing vectors, comprising the steps of:

loading a vector;
accessing arbitrary portions of the vector; and
5 performing a specified operation using the
accessed portions of the vector.

35. The method of claim 34, further comprising:

obtaining pointer information;
10 identifying the portions of the vector to access
using the pointer information;
accessing the identified portions of the vector;
and
updating updating the pointer information to
15 enable a sequential read of a vector file.

36. The method of claim 34, further comprising:

obtaining pointer information;
identifying the portions of the vector to access
20 using the pointer information;
accessing the identified portions of the vector;
and

updating the pointer information to enable a
non-sequential read of a vector file.

37. The method of claim 35, further comprising:
5 obtaining pointer information;
identifying the portions of the vector to access
using the pointer information;
accessing the identified portions of the vector;
and
10 updating the pointer information to enable a
sequential write to a vector file.

38. The method of claim 35, further comprising:
obtaining pointer information;
15 identifying the portions of the vector to access
using the pointer information;
accessing the identified portions of the vector;
and
updating the pointer information to enable a
20 non-sequential write to a vector file.

39. The method of claim 34, further comprising:
obtaining pointer information;
identifying the portions of the vector to access
using the pointer information;
5 accessing the identified portions of the vector;
and
updating the pointer information to enable an
indirect read of a vector file.

10 40. The method of claim 34, further comprising:
obtaining pointer information;
identifying the portions of the vector to access
using the pointer information;
accessing the identified portions of the vector;
15 and
updating the pointer information to enable an
indirect write to a vector file.

20 41. The method of claim 34, further comprising:
obtaining pointer information;
identifying the portions of the vector to access
using the pointer information;

accessing the identified portions of the vector;
and
updating the pointer information by a stride
value.

5

42. The method of claim 34, further comprising:
obtaining pointer information;
identifying the portions of the vector to access
using the pointer information;
10 accessing the identified portions of the vector;
and
updating the pointer information to enable modulo
addressing accesses to a vector file.

15

43. A program storage device readable by a machine,
tangibly embodying a program of instructions executable on
the machine to perform method steps for processing vectors,
the method steps comprising:

20

loading a vector;
accessing arbitrary portions of the vector; and
performing a specified operation using the
accessed portions of the vector.

44. A method for identifying portions of a vector to be processed, comprising the step of loading vector boundary information for accessing portions of the vector.

5

45. The method of claim 44, wherein the vector boundary information comprises arbitrary vector addresses.

10

46. The method of claim 45, wherein the step of loading the vector boundary information comprises executing a program instruction specifying the vector boundary information.

15

47. The method of claim 46, wherein the program instruction is a load instruction (VPTRLOAD).

48. A system for facilitating processing of vectors, comprising:

20

a vector memory area; and
a controller for performing a specified operation on arbitrary portions of a vector stored in the vector memory area.

49. The system of claim 48, further comprising a pointer memory area containing address information identifying the portions of the vector to access.

5 50. The system of claim 48, wherein the address information is loaded into the pointer memory area by a program instruction.

10 51. The system of claim 50, wherein the program instruction is a load instruction (VPTRLOAD).

15 52. The system of claim 49, wherein the address information stored in the pointer memory area comprises a plurality of pointers pointing to a portion of the vector memory area.

20 53. The system of claim 52, wherein the plurality of pointers are stored in the pointer memory area as a pointer array.

54. The system of claim 53, wherein each entry of the pointer array includes an address representing the row and column of at least one element in the vector memory area.

5 55. The system of claim 48, wherein the vector memory area is organized in a matrix of rows and columns.

56. The system of claim 48, wherein the vector memory area is organized in a linear array.

10

57. The system of claim 53, wherein the pointer array is organized in a matrix of rows and columns.

58. A system for processing operations that use data
15 vectors each comprising a plurality of data elements comprising:

a vector data file comprising a plurality of storage elements for storing data elements of the data vectors;

20 a pointer array coupled by a bus to the vector data file, the pointer array including a plurality of

entries wherein each entry identifies at least one storage element in the vector data file; and

the at least one storage element for storing at least one data element of the data vectors, wherein for at least one particular entry in the pointer array, the at least one storage element identified by the particular entry has an arbitrary starting address in the vector data file.

59. A method for processing operations that use data vectors each comprising a plurality of data elements, the method comprising the steps of:

providing a vector data file comprising a plurality of storage elements for storing data elements of the data vectors, and

providing a pointer array having a plurality of entries, wherein each entry identifies at least one storage element in the vector data file for storing at least one data element of the data vectors, wherein for at least one particular entry in the pointer array, the at least one storage element identified by the particular entry has an arbitrary starting address in the vector data file.

60. A program storage device readable by machine,
tangibly embodying a program of instructions executable by
the machine to perform method steps for processing
operations that use data vectors each comprising a plurality
of data elements, the method steps comprising:

providing a vector data file comprising a
plurality of storage elements for storing data elements of
the data vectors, and

providing a pointer array having a plurality of
entries, wherein each entry identifies at least one storage
element in the vector data file for storing at least one
data element of the data vectors, wherein for at least one
particular entry in the pointer array, the at least one
storage element identified by the particular entry has an
arbitrary starting address in the vector data file.